

Mastering Embedded System Online Diploma

WWW.learn-in-depth.com

Report

Pressure Controller

First Term (Final Project 1)

Eng. Mohamed Abdallah Tawfik

My profile : <https://www.learn-in-depth.com/online-diploma/mohamedabdallahtawfik14@gmail.com>

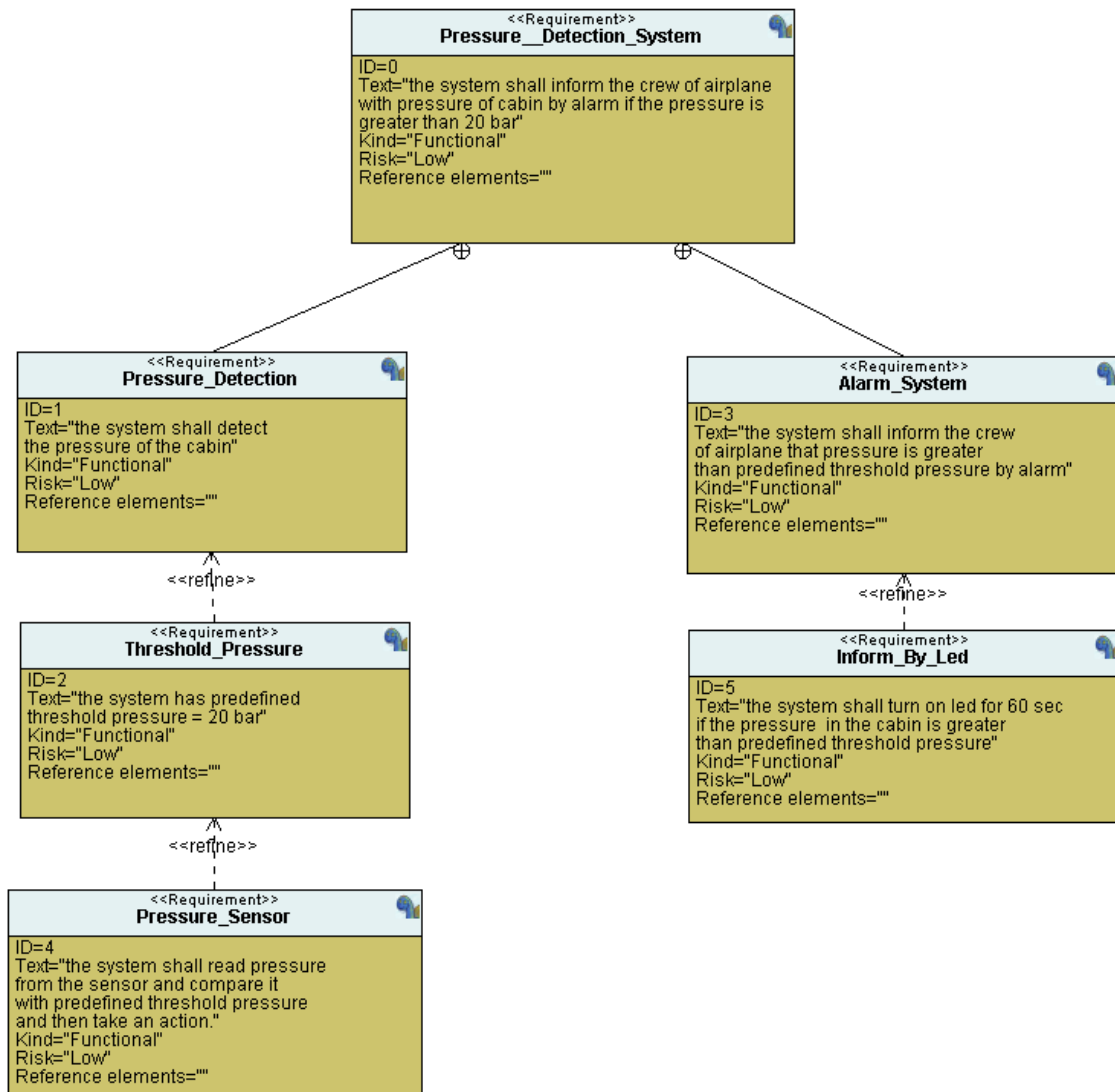
Case Study:

Pressure system to inform crew of airplane that pressure is greater than 20 bar by alarm (turn on led for 60 sec).

Assumptions:

1. Pressure sensor never fails.
2. The system maintenance is not modeled.
3. Alarm system never fails.

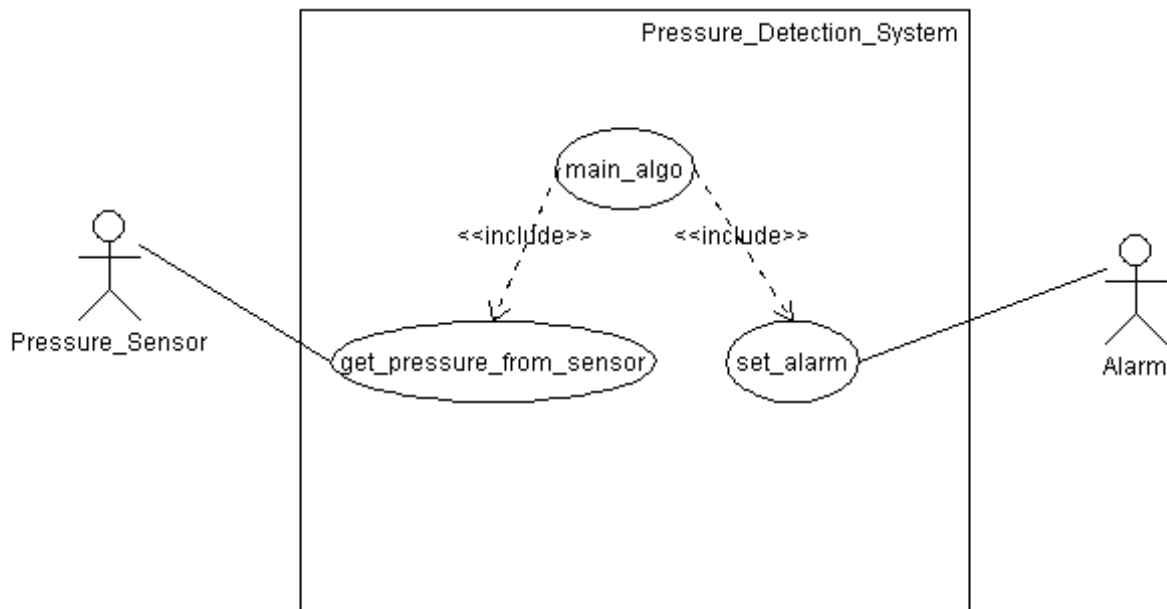
Requirements:



System Analysis:

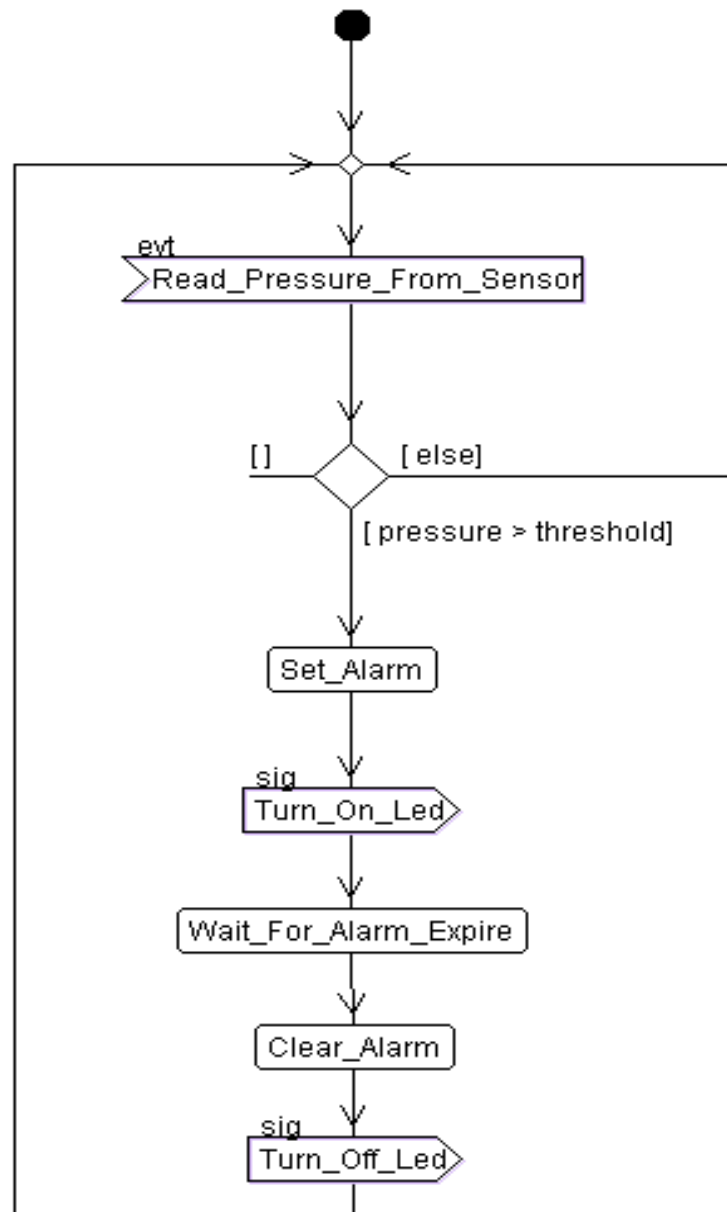
- **Use case diagram**

- 1. Get_pressure_from_sensor will get pressure value from Presure_Sensor actor.**
- 2. Set_alarm will turn on led for 60 secs if the pressure which we got on it from sensor was greater than 20 bar.**



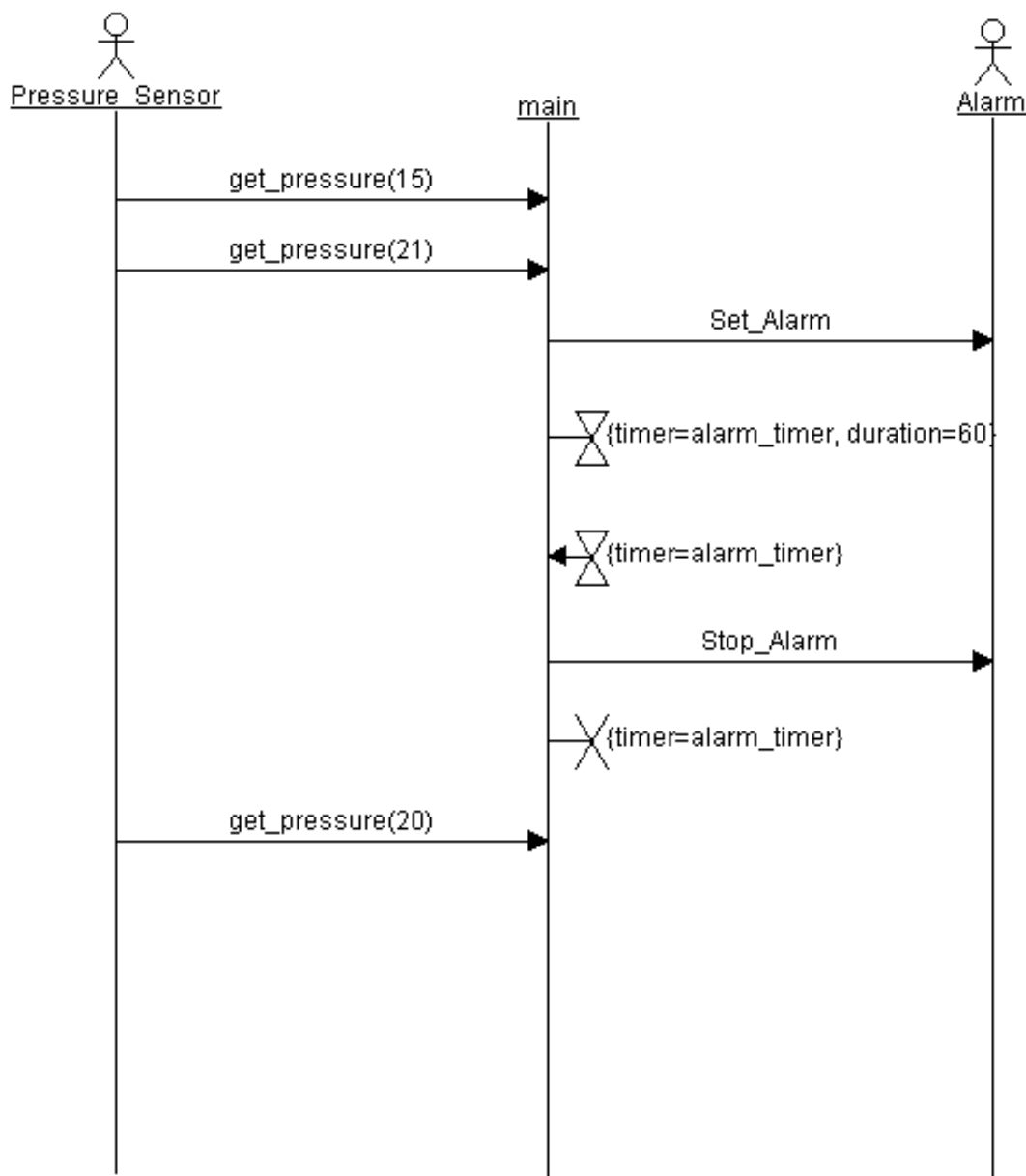
- **Activity diagram**

- read from pressure sensor then take an action based on that, if the pressure measured via sensor is greater than threshold will set alarm and send signal to the led to make it turning on and wait for 60 secs then clear alarm and send signal to led to make it turning off again.



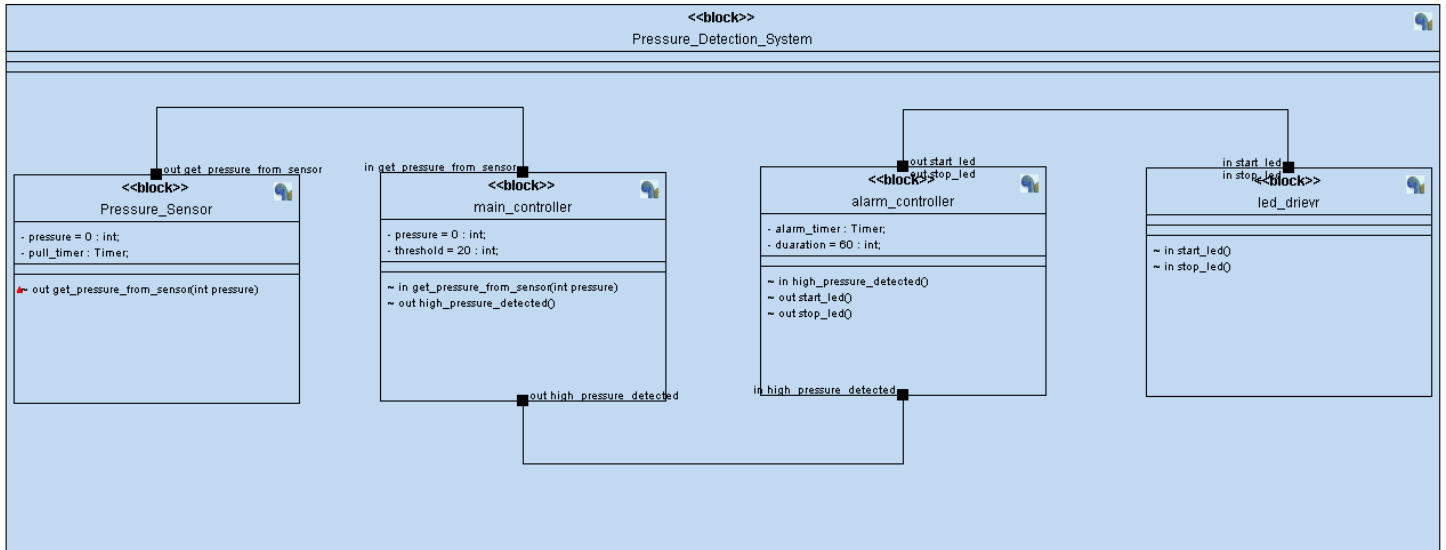
- **Sequence diagram example**

- sensor measured pressure = 15 bar less than threshold 20 bar so the system takes no action.
- then the sensor measure pressure again = 21 greater than threshold 20 bar so the system send signal to alarm to start and turn the led on and wait for 60 secs after expiration of the timer it sends signal again to stop alarm on turn the led off then clear the timer.
- Sensor sends new measure for pressure = 20 bar and it's not greater than the threshold 20 bar so the system takes no action.

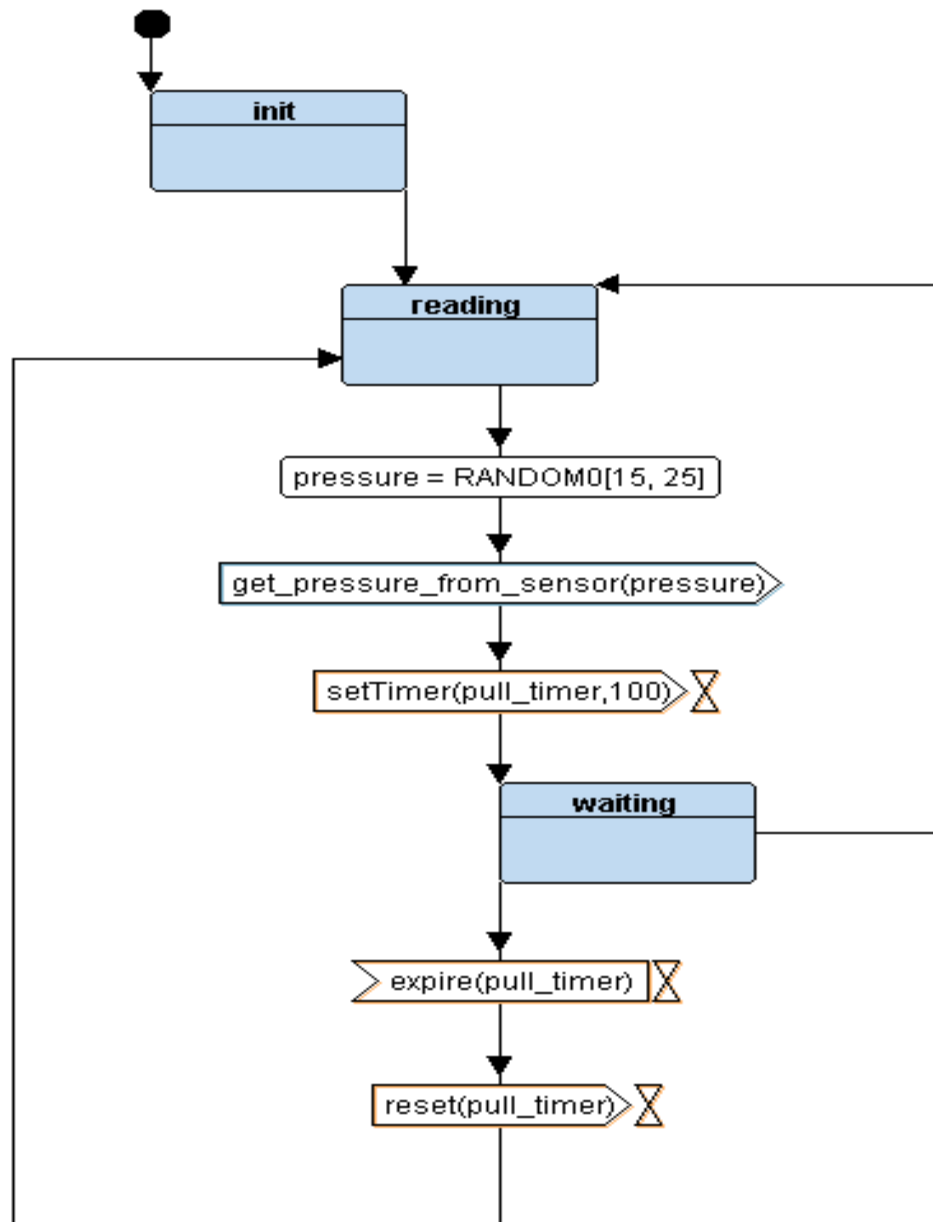


System Design:

• Block design

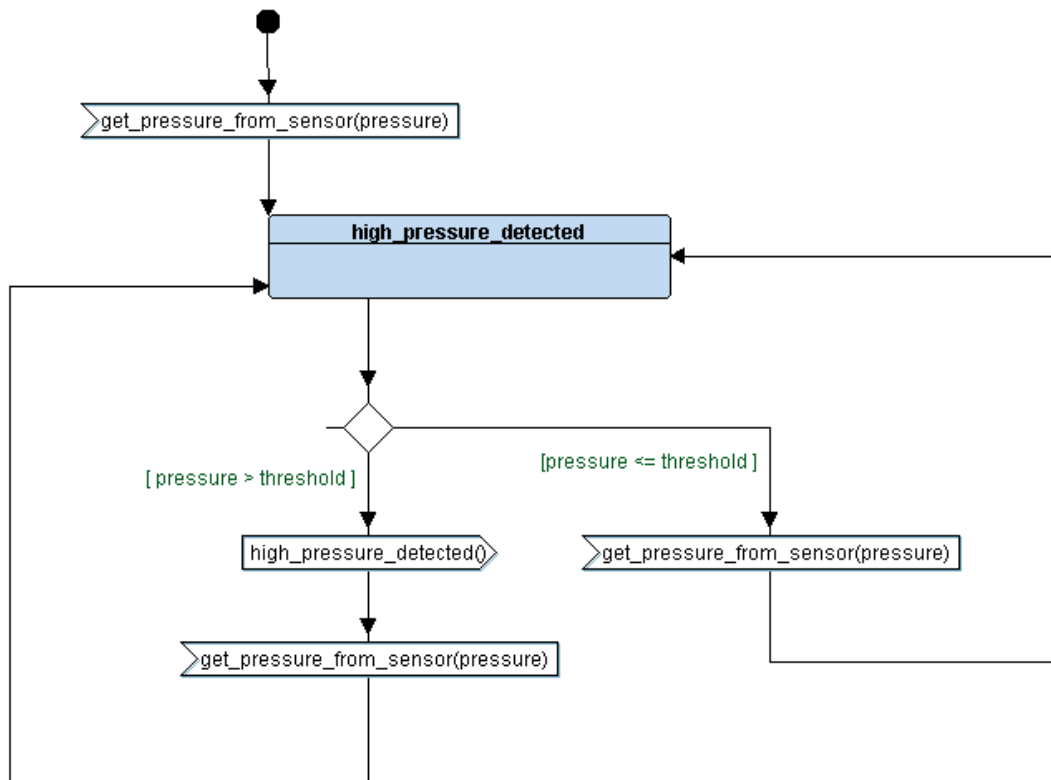


- **Pressure Sensor state**



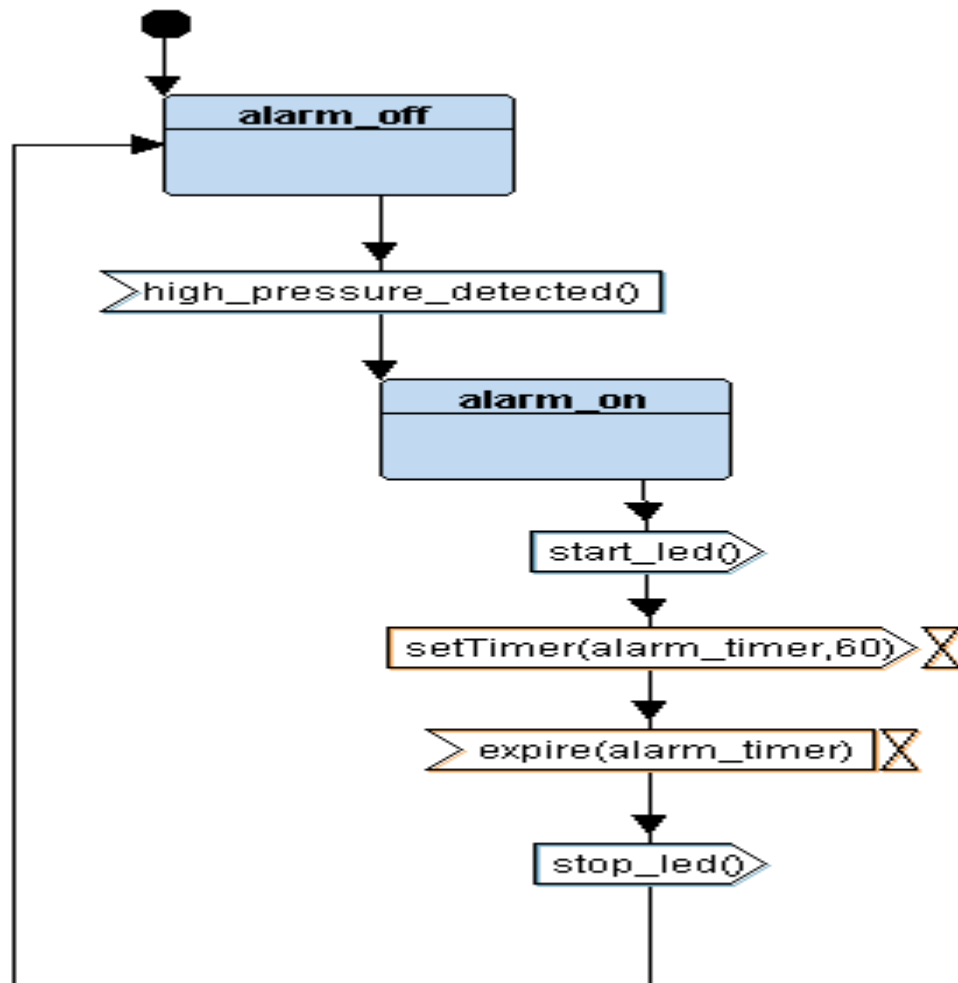
- First initialize pressure sensor then turn in reading state, then send the sensor measure to main controller and wait for 100 secs then go to reading state again.

- **Main controller state**



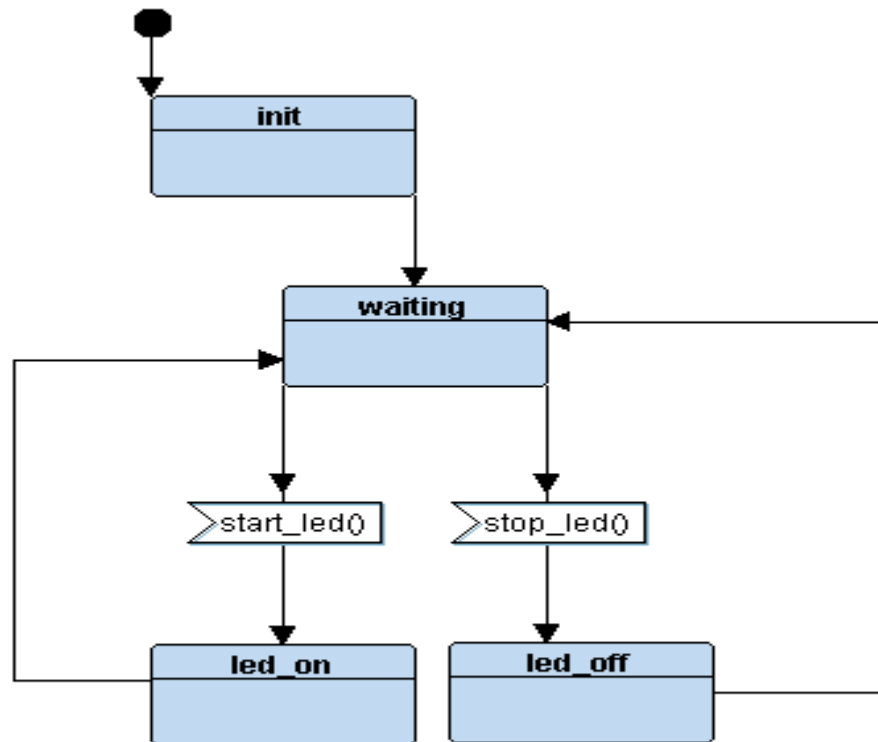
- Received measure of sensor then make decision based on the predefined threshold if pressure measured by the sensor was larger than threshold will send signal to alarm driver.

- Alarm driver state



- First alarm will be off then when he received a signal from main controller he will send signal to led driver and set the timer for 60 secs and after 60 secs he will send signal to led driver again to make it off then go to off mode again and wait for new signal.

- **Led driver state**



- First initialize led driver then waiting for a signal from alarm controller, if it received a start it will go to led on else will turn off led.

- **Symbols and sections**

Symbols for pressure system

```
MINGW64:/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-nm.exe pressureSystem.elf
0800019c t _reset
080001a2 t _vector_handler
0800004c T Delay
08000070 T getPressureVal
080000d8 T GPIO_INITIALIZATION
0800001c T high_pressure_detected
08000158 T main
20000000 B pressure
08000088 T Set_Alarm_actuator
080001a4 D threshold

MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$
```

Symbols for main controller

```
MINGW64:/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-nm.exe main.o
00000000 U Delay
00000000 U getPressureVal
00000000 U GPIO_INITIALIZATION
00000000 U high_pressure_detected
00000000 T main
00000004 C pressure
00000000 D threshold
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ |
```

Symbols for alarm driver

```
MINGW64:/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-nm.exe alarm.o
00000000 U Delay
00000000 T high_pressure_detected
00000000 U Set_Alarm_actuator
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ |
```

Symbols for led driver

```
MINGW64:/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-nm.exe driver.o
00000000 T Delay
00000024 T getPressureVal
0000008c T GPIO_INITIALIZATION
0000003c T Set_Alarm_actuator

MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$
```

- **Sections**

Sections for pressure system

```
MINGW64:/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-objdump.exe -h pressureSystem.elf

pressureSystem.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          000001a4  08000000  08000000  00008000  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data          00000004  080001a4  080001a4  000081a4  2**2
    CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000004  20000000  20000000  00010000  2**2
    ALLOC
 3 .debug_info     000002b6  00000000  00000000  000081a8  2**0
    CONTENTS, READONLY, DEBUGGING
 4 .debug_abbrev   0000016d  00000000  00000000  0000845e  2**0
    CONTENTS, READONLY, DEBUGGING
 5 .debug_loc      00000120  00000000  00000000  000085cb  2**0
    CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges  00000080  00000000  00000000  000086f0  2**3
    CONTENTS, READONLY, DEBUGGING
 7 .debug_line     00000148  00000000  00000000  00008770  2**0
    CONTENTS, READONLY, DEBUGGING
 8 .debug_str      00000153  00000000  00000000  000088b8  2**0
    CONTENTS, READONLY, DEBUGGING
 9 .comment        00000011  00000000  00000000  00008a0b  2**0
    CONTENTS, READONLY
10 .ARM.attributes 00000031  00000000  00000000  00008a1c  2**0
    CONTENTS, READONLY
11 .debug_frame    000000d0  00000000  00000000  00008a50  2**2
    CONTENTS, READONLY, DEBUGGING
```

Sections for alarm driver

```
MINGW64:/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-objdump.exe -h alarm.o

alarm.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000030 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data          00000000 00000000 00000000 00000064 2**0
CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000000 00000000 00000000 00000064 2**0
ALLOC
 3 .debug_info    0000006a 00000000 00000000 00000064 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev  00000062 00000000 00000000 000000ce 2**0
CONTENTS, READONLY, DEBUGGING
 5 .debug_loc     0000002c 00000000 00000000 00000130 2**0
CONTENTS, READONLY, DEBUGGING
 6 .debug_ranges  00000020 00000000 00000000 0000015c 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line    0000003a 00000000 00000000 0000017c 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str     0000009f 00000000 00000000 000001b6 2**0
CONTENTS, READONLY, DEBUGGING
 9 .comment       00000012 00000000 00000000 00000255 2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 00000267 2**0
CONTENTS, READONLY
11 .debug_frame   0000002c 00000000 00000000 0000029c 2**2
CONTENTS, RELOC, READONLY, DEBUGGING

MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$
```

Sections main controller

```
MINGW64:/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-objdump.exe -h main.o

main.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000044 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data          00000004 00000000 00000000 00000078 2**2
CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000000 00000000 00000000 0000007c 2**0
ALLOC
 3 .debug_info    000000bc 00000000 00000000 0000007c 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev  0000005a 00000000 00000000 00000138 2**0
CONTENTS, READONLY, DEBUGGING
 5 .debug_loc     0000002c 00000000 00000000 00000192 2**0
CONTENTS, READONLY, DEBUGGING
 6 .debug_ranges  00000020 00000000 00000000 000001be 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line    0000003a 00000000 00000000 000001de 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str     00000118 00000000 00000000 00000218 2**0
CONTENTS, READONLY, DEBUGGING
 9 .comment       00000012 00000000 00000000 00000330 2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 00000342 2**0
CONTENTS, READONLY
11 .debug_frame   0000002c 00000000 00000000 00000378 2**2
CONTENTS, RELOC, READONLY, DEBUGGING

MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$
```

Sections for led driver

```

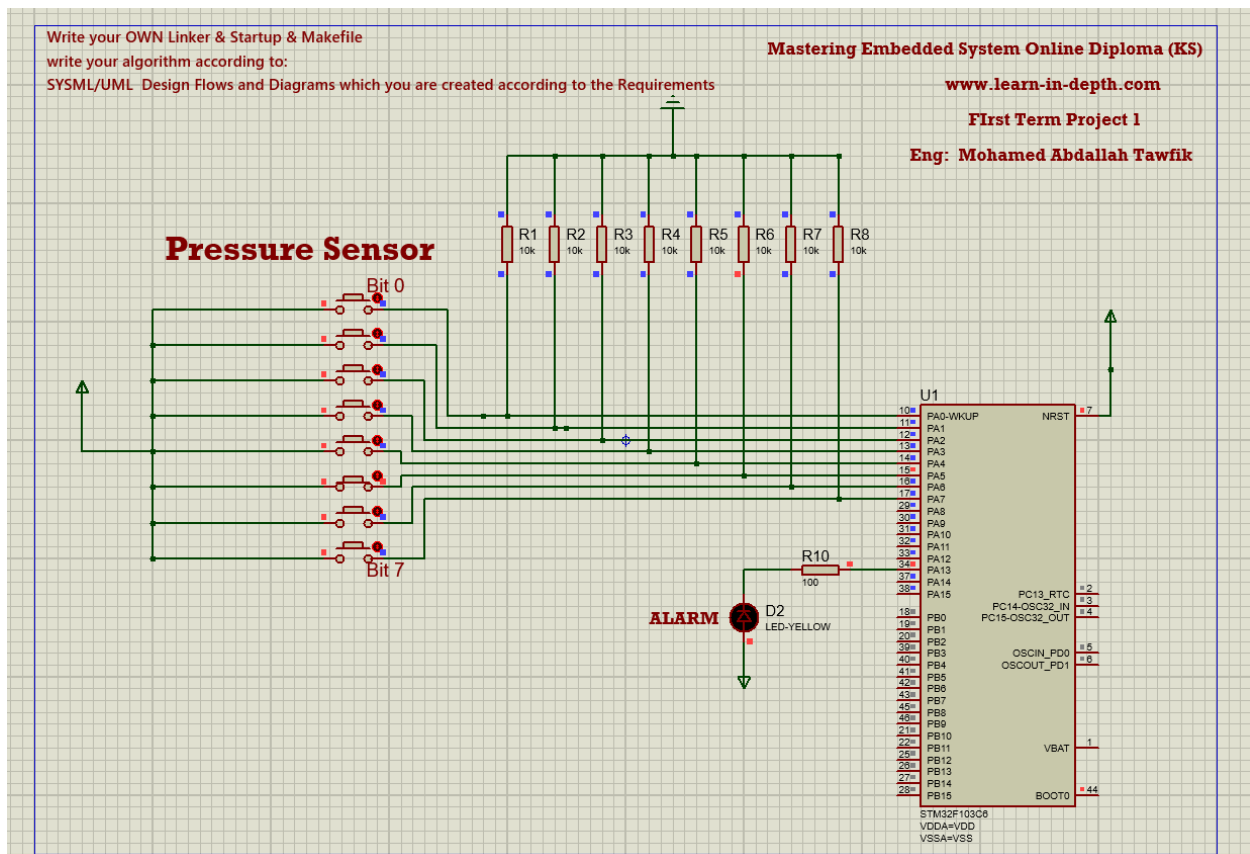
MINGW64/e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code
MoStudy@DESKTOP-U1MGQM3 MINGW64 /e/Abdallah/Master/Embedded-System-Online-Diploma/projects/first_term/final_project 1/code (main)
$ arm-none-eabi-objdump.exe -h driver.o

driver.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          0000010c  00000000  00000000  00000034  2**2
CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data           00000000  00000000  00000000  00000140  2**0
CONTENTS, ALLOC, LOAD, DATA
  2 .bss            00000000  00000000  00000000  00000140  2**0
ALLOC
  3 .debug_info     00000103  00000000  00000000  00000140  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev   0000009d  00000000  00000000  00000243  2**0
CONTENTS, READONLY, DEBUGGING
  5 .debug_loc      000000c8  00000000  00000000  000002e0  2**0
CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges  00000020  00000000  00000000  000003a8  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line     00000099  00000000  00000000  000003c8  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str      0000014e  00000000  00000000  00000461  2**0
CONTENTS, READONLY, DEBUGGING
  9 .comment        00000012  00000000  00000000  000005af  2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033  00000000  00000000  000005c1  2**0
CONTENTS, READONLY
11 .debug_frame     00000078  00000000  00000000  000005f4  2**2
CONTENTS, RELOC, READONLY, DEBUGGING

```

- When pressure larger than 20 (32)



• Less than 20 (8)

